# FINAL PACKAGE

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

struct Voter {

string id;

string name;

string password;

bool hasVoted;

};

struct Candidate {

string id;

string name;

string party;

int votes;

};

vector<Voter> voters;

vector<Candidate> candidates;

const string ADMIN\_USER = "admin";

const string ADMIN\_PASS = "admin123";

void loadData() {

// Load voters

ifstream voterFile("Votersdetails.txt");

if (voterFile) {

Voter v;

string voted;

while (voterFile >> v.id >> v.name >> v.password >> voted) {

v.hasVoted = (voted == "1");

voters.push\_back(v);

}

voterFile.close();

}

// Load candidates

ifstream candidateFile("candidates.txt");

if (candidateFile) {

Candidate c;

while (candidateFile >> c.id >> c.name >> c.party >> c.votes) {

candidates.push\_back(c);

}

candidateFile.close();

}

}

void saveData() {

// Save voters

ofstream voterFile("Votersdetails.txt");

for (const auto& v : voters) {

voterFile << v.id << " " << v.name << " " << v.password

<< " " << (v.hasVoted ? "1" : "0") << "\n";

}

voterFile.close();

// Save candidates

ofstream candidateFile("candidates.txt");

for (const auto& c : candidates) {

candidateFile << c.id << " " << c.name << " " << c.party

<< " " << c.votes << "\n";

}

candidateFile.close();

}

bool voterLogin() {

string id, password;

cout << "Enter Voter ID: ";

cin >> id;

cout << "Enter Password: ";

cin >> password;

for (auto& v : voters) {

if (v.id == id && v.password == password) {

if (v.hasVoted) {

cout << "You have already voted!\n";

return false;

}

return true;

}

}

cout << "Invalid credentials!\n";

return false;

}

void displayCandidates() {

cout << "\nAvailable Candidates:\n";

cout << "ID\tName\t\tParty\n";

for (const auto& c : candidates) {

cout << c.id << "\t" << c.name << "\t" << c.party << "\n";

}

}

void castVote(const string& voterId) {

displayCandidates();

string candidateId;

cout << "Enter candidate ID to vote: ";

cin >> candidateId;

auto it = find\_if(candidates.begin(), candidates.end(),

[&](const Candidate& c) { return c.id == candidateId; });

if (it != candidates.end()) {

it->votes++;

for (auto& v : voters) {

if (v.id == voterId) {

v.hasVoted = true;

break;

}

}

saveData();

cout << "Vote cast successfully!\n";

} else {

cout << "Invalid candidate ID!\n";

}

}

void adminView() {

cout << "\nVoter List:\n";

for (const auto& v : voters) {

cout << v.id << "\t" << v.name << "\t"

<< (v.hasVoted ? "Voted" : "Not Voted") << "\n";

}

cout << "\nCandidate Results:\n";

for (const auto& c : candidates) {

cout << c.name << " (" << c.party << "): " << c.votes << " votes\n";

}

}

int main() {

loadData();

while (true) {

cout << "\n1. Voter Login\n2. Admin Login\n3. Exit\nChoice: ";

int choice;

cin >> choice;

if (choice == 1) {

if (voterLogin()) {

string voterId;

cout << "Enter your voter ID again: ";

cin >> voterId;

castVote(voterId);

}

}

else if (choice == 2) {

string user, pass;

cout << "Admin Username: ";

cin >> user;

cout << "Admin Password: ";

cin >> pass;

if (user == ADMIN\_USER && pass == ADMIN\_PASS) {

adminView();

} else {

cout << "Invalid admin credentials!\n";

}

}

else if (choice == 3) {

saveData();

cout << "Exiting system...\n";

break;

}

else {

cout << "Invalid choice!\n";

}

}

return 0;

}

HTML CODE:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Online Voting System</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background-color: #f5f5f5;

margin: 0;

padding: 0;

color: #333;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

header {

background-color: #2c3e50;

color: white;

padding: 20px 0;

text-align: center;

margin-bottom: 30px;

box-shadow: 0 2px 5px rgba(0,0,0,0.1);

}

h1 {

margin: 0;

}

.card {

background-color: white;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

padding: 25px;

margin-bottom: 30px;

}

.form-group {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 8px;

font-weight: 600;

}

input[type="text"], input[type="password"], input[type="number"], select {

width: 100%;

padding: 12px;

border: 1px solid #ddd;

border-radius: 4px;

font-size: 16px;

box-sizing: border-box;

}

button {

background-color: #3498db;

color: white;

border: none;

padding: 12px 20px;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s;

}

button:hover {

background-color: #2980b9;

}

.btn-danger {

background-color: #e74c3c;

}

.btn-danger:hover {

background-color: #c0392b;

}

.btn-success {

background-color: #2ecc71;

}

.btn-success:hover {

background-color: #27ae60;

}

.hidden {

display: none;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

padding: 12px 15px;

text-align: left;

border-bottom: 1px solid #ddd;

}

th {

background-color: #f2f2f2;

font-weight: 600;

}

tr:hover {

background-color: #f9f9f9;

}

.nav-tabs {

display: flex;

border-bottom: 1px solid #ddd;

margin-bottom: 20px;

}

.tab {

padding: 10px 20px;

cursor: pointer;

background-color: #f1f1f1;

border: 1px solid #ddd;

border-bottom: none;

margin-right: 5px;

border-radius: 5px 5px 0 0;

}

.tab.active {

background-color: white;

border-bottom: 1px solid white;

margin-bottom: -1px;

}

.tab-content {

display: none;

}

.tab-content.active {

display: block;

}

.vote-btn {

padding: 6px 12px;

background-color: #3498db;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

.logout-btn {

background-color: #7f8c8d;

float: right;

}

.logout-btn:hover {

background-color: #95a5a6;

}

.result-bar {

height: 20px;

background-color: #3498db;

margin-top: 5px;

border-radius: 3px;

}

.candidate-card {

display: flex;

align-items: center;

padding: 15px;

border: 1px solid #ddd;

border-radius: 5px;

margin-bottom: 15px;

background-color: #fff;

}

.candidate-info {

flex-grow: 1;

}

.candidate-name {

font-weight: bold;

font-size: 18px;

margin-bottom: 5px;

}

.candidate-party {

color: #666;

}

.candidate-votes {

font-weight: bold;

color: #2c3e50;

}

</style>

</head>

<body>

<header>

<div class="container">

<h1>Online Voting System</h1>

</div>

</header>

<div class="container">

<!-- Login Section -->

<div id="login-section" class="card">

<h2>Login</h2>

<div class="form-group">

<label for="login-type">Login As:</label>

<select id="login-type" class="form-control">

<option value="voter">Voter</option>

<option value="admin">Admin</option>

</select>

</div>

<div class="form-group">

<label for="user-id">User ID:</label>

<input type="text" id="user-id" placeholder="Enter your ID">

</div>

<div class="form-group">

<label for="password">Password:</label>

<input type="password" id="password" placeholder="Enter your password">

</div>

<button id="login-btn" class="btn-success">Login</button>

</div>

<!-- Voter Dashboard -->

<div id="voter-dashboard" class="card hidden">

<h2>Welcome, <span id="voter-name"></span></h2>

<button id="voter-logout" class="logout-btn">Logout</button>

<div class="nav-tabs">

<div class="tab active" data-tab="vote">Cast Vote</div>

<div class="tab" data-tab="instructions">Instructions</div>

</div>

<div id="vote-tab" class="tab-content active">

<h3>Available Candidates</h3>

<div id="candidates-list"></div>

</div>

<div id="instructions-tab" class="tab-content">

<h3>Voting Instructions</h3>

<ol>

<li>Review the list of candidates carefully.</li>

<li>Select the candidate you wish to vote for by clicking the "Vote" button.</li>

<li>Once you cast your vote, you cannot change it.</li>

<li>Each voter is allowed to vote only once.</li>

<li>The voting process is anonymous and secure.</li>

</ol>

</div>

</div>

<!-- Admin Dashboard -->

<div id="admin-dashboard" class="card hidden">

<h2>Admin Dashboard</h2>

<button id="admin-logout" class="logout-btn">Logout</button>

<div class="nav-tabs">

<div class="tab active" data-tab="voters">Voter List</div>

<div class="tab" data-tab="results">Election Results</div>

<div class="tab" data-tab="manage">Manage System</div>

</div>

<div id="voters-tab" class="tab-content active">

<h3>Registered Voters</h3>

<table id="voters-table">

<thead>

<tr>

<th>Voter ID</th>

<th>Name</th>

<th>Age</th>

<th>Role</th>

<th>Voting Status</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

<div id="results-tab" class="tab-content">

<h3>Election Results</h3>

<div id="results-chart"></div>

</div>

<div id="manage-tab" class="tab-content">

<h3>System Management</h3>

<div class="form-group">

<button id="add-voter-btn" class="btn-success">Add New Voter</button>

</div>

<div class="form-group">

<button id="add-candidate-btn" class="btn-success">Add New Candidate</button>

</div>

<div class="form-group">

<button id="reset-system-btn" class="btn-danger">Reset Voting System</button>

</div>

</div>

</div>

<!-- Add Voter Modal -->

<div id="add-voter-modal" class="card hidden">

<h3>Add New Voter</h3>

<div class="form-group">

<label for="new-voter-id">Voter ID:</label>

<input type="text" id="new-voter-id" placeholder="Enter voter ID">

</div>

<div class="form-group">

<label for="new-voter-name">Full Name:</label>

<input type="text" id="new-voter-name" placeholder="Enter full name">

</div>

<div class="form-group">

<label for="new-voter-age">Age:</label>

<input type="number" id="new-voter-age" placeholder="Enter age">

</div>

<div class="form-group">

<label for="new-voter-password">Password:</label>

<input type="password" id="new-voter-password" placeholder="Enter password">

</div>

<button id="save-voter-btn" class="btn-success">Save Voter</button>

<button id="cancel-voter-btn" class="btn-danger">Cancel</button>

</div>

<!-- Add Candidate Modal -->

<div id="add-candidate-modal" class="card hidden">

<h3>Add New Candidate</h3>

<div class="form-group">

<label for="new-candidate-id">Candidate ID:</label>

<input type="text" id="new-candidate-id" placeholder="Enter candidate ID">

</div>

<div class="form-group">

<label for="new-candidate-name">Full Name:</label>

<input type="text" id="new-candidate-name" placeholder="Enter full name">

</div>

<div class="form-group">

<label for="new-candidate-party">Political Party:</label>

<input type="text" id="new-candidate-party" placeholder="Enter party affiliation">

</div>

<button id="save-candidate-btn" class="btn-success">Save Candidate</button>

<button id="cancel-candidate-btn" class="btn-danger">Cancel</button>

</div>

</div>

<script>

// Updated voter data

let voters = [

{ id: "IND01", name: "ANITHA", age: 39, password: "IND001", role: "Voter", hasVoted: false },

{ id: "IND02", name: "ANU", age: 23, password: "IND002", role: "Voter", hasVoted: false },

{ id: "IND03", name: "PRABHA", age: 39, password: "IND003", role: "Voter", hasVoted: false },

{ id: "IND04", name: "PRIYA", age: 44, password: "IND004", role: "Voter", hasVoted: false },

{ id: "IND05", name: "INDIRA", age: 48, password: "IND005", role: "Voter", hasVoted: false },

{ id: "IND06", name: "SETHU", age: 21, password: "IND007", role: "Voter", hasVoted: false },

{ id: "IND07", name: "PREM", age: 27, password: "IND008", role: "Voter", hasVoted: false },

{ id: "IND08", name: "AJAY", age: 23, password: "IND123", role: "Voter", hasVoted: false },

{ id: "IND09", name: "VISWA", age: 36, password: "IND231", role: "Voter", hasVoted: false },

{ id: "IND10", name: "AMITHA", age: 18, password: "IND982", role: "Voter", hasVoted: false }

];

// Updated candidate data

let candidates = [

{ id: "C001", name: "Preetha", party: "DMK", votes: 0 },

{ id: "C002", name: "Vijay", party: "TVK", votes: 0 }

];

const ADMIN\_USER = "admin";

const ADMIN\_PASS = "admin123";

// DOM Elements

const loginSection = document.getElementById('login-section');

const voterDashboard = document.getElementById('voter-dashboard');

const adminDashboard = document.getElementById('admin-dashboard');

const voterNameSpan = document.getElementById('voter-name');

const candidatesList = document.getElementById('candidates-list');

const votersTable = document.getElementById('voters-table').querySelector('tbody');

const resultsChart = document.getElementById('results-chart');

// Modal elements

const addVoterModal = document.getElementById('add-voter-modal');

const addCandidateModal = document.getElementById('add-candidate-modal');

// Current user

let currentUser = null;

// Initialize the app

document.addEventListener('DOMContentLoaded', function() {

setupEventListeners();

// In a real app, you would load data from the backend here

// loadData();

});

function setupEventListeners() {

// Login button

document.getElementById('login-btn').addEventListener('click', handleLogin);

// Logout buttons

document.getElementById('voter-logout').addEventListener('click', logout);

document.getElementById('admin-logout').addEventListener('click', logout);

// Tab navigation

document.querySelectorAll('.tab').forEach(tab => {

tab.addEventListener('click', function() {

const tabId = this.getAttribute('data-tab');

switchTab(this, tabId);

});

});

// Admin management buttons

document.getElementById('add-voter-btn').addEventListener('click', () => {

addVoterModal.classList.remove('hidden');

});

document.getElementById('add-candidate-btn').addEventListener('click', () => {

addCandidateModal.classList.remove('hidden');

});

document.getElementById('reset-system-btn').addEventListener('click', resetSystem);

// Modal buttons

document.getElementById('cancel-voter-btn').addEventListener('click', () => {

addVoterModal.classList.add('hidden');

});

document.getElementById('cancel-candidate-btn').addEventListener('click', () => {

addCandidateModal.classList.add('hidden');

});

document.getElementById('save-voter-btn').addEventListener('click', saveNewVoter);

document.getElementById('save-candidate-btn').addEventListener('click', saveNewCandidate);

}

function handleLogin() {

const loginType = document.getElementById('login-type').value;

const userId = document.getElementById('user-id').value;

const password = document.getElementById('password').value;

if (loginType === 'voter') {

const voter = voters.find(v => v.id === userId && v.password === password);

if (voter) {

if (voter.hasVoted) {

alert('You have already voted!');

return;

}

currentUser = voter;

showVoterDashboard();

} else {

alert('Invalid credentials!');

}

} else if (loginType === 'admin') {

if (userId === ADMIN\_USER && password === ADMIN\_PASS) {

currentUser = { id: 'admin', name: 'Administrator' };

showAdminDashboard();

} else {

alert('Invalid admin credentials!');

}

}

}

function showVoterDashboard() {

loginSection.classList.add('hidden');

voterDashboard.classList.remove('hidden');

voterNameSpan.textContent = currentUser.name;

renderCandidates();

}

function showAdminDashboard() {

loginSection.classList.add('hidden');

adminDashboard.classList.remove('hidden');

renderVotersTable();

renderResults();

}

function renderCandidates() {

candidatesList.innerHTML = '';

candidates.forEach(candidate => {

const card = document.createElement('div');

card.className = 'candidate-card';

card.innerHTML = `

<div class="candidate-info">

<div class="candidate-name">${candidate.name}</div>

<div class="candidate-party">${candidate.party}</div>

</div>

<button class="vote-btn" data-id="${candidate.id}">Vote</button>

`;

candidatesList.appendChild(card);

});

// Add event listeners to vote buttons

document.querySelectorAll('.vote-btn').forEach(btn => {

btn.addEventListener('click', function() {

const candidateId = this.getAttribute('data-id');

castVote(candidateId);

});

});

}

function renderVotersTable() {

votersTable.innerHTML = '';

voters.forEach(voter => {

const row = document.createElement('tr');

row.innerHTML = `

<td>${voter.id}</td>

<td>${voter.name}</td>

<td>${voter.age}</td>

<td>${voter.role}</td>

<td>${voter.hasVoted ? 'Voted' : 'Not Voted'}</td>

`;

votersTable.appendChild(row);

});

}

function renderResults() {

resultsChart.innerHTML = '';

// Sort candidates by votes (descending)

const sortedCandidates = [...candidates].sort((a, b) => b.votes - a.votes);

sortedCandidates.forEach(candidate => {

const totalVotes = voters.filter(v => v.hasVoted).length;

const percentage = totalVotes > 0 ? Math.round((candidate.votes / totalVotes) \* 100) : 0;

const resultDiv = document.createElement('div');

resultDiv.className = 'result-item';

resultDiv.style.marginBottom = '15px';

resultDiv.innerHTML = `

<div style="display: flex; justify-content: space-between;">

<span><strong>${candidate.name}</strong> (${candidate.party})</span>

<span>${candidate.votes} votes (${percentage}%)</span>

</div>

<div class="result-bar" style="width: ${percentage}%"></div>

`;

resultsChart.appendChild(resultDiv);

});

}

function castVote(candidateId) {

if (confirm('Are you sure you want to vote for this candidate? This action cannot be undone.')) {

// Find the candidate

const candidate = candidates.find(c => c.id === candidateId);

if (candidate) {

// Update candidate votes

candidate.votes++;

// Update voter status

const voter = voters.find(v => v.id === currentUser.id);

if (voter) {

voter.hasVoted = true;

}

// Save changes (in a real app, this would be an API call)

// saveData();

// Show success message

alert('Thank you for voting! Your vote has been recorded.');

// Logout the voter since they can't vote again

logout();

}

}

}

function saveNewVoter() {

const id = document.getElementById('new-voter-id').value;

const name = document.getElementById('new-voter-name').value;

const age = document.getElementById('new-voter-age').value;

const password = document.getElementById('new-voter-password').value;

if (!id || !name || !age || !password) {

alert('Please fill in all fields');

return;

}

// Add new voter

voters.push({

id,

name,

age: parseInt(age),

password,

role: "Voter",

hasVoted: false

});

// Clear form

document.getElementById('new-voter-id').value = '';

document.getElementById('new-voter-name').value = '';

document.getElementById('new-voter-age').value = '';

document.getElementById('new-voter-password').value = '';

// Hide modal

addVoterModal.classList.add('hidden');

// Refresh voter table

renderVotersTable();

alert('New voter added successfully!');

}

function saveNewCandidate() {

const id = document.getElementById('new-candidate-id').value;

const name = document.getElementById('new-candidate-name').value;

const party = document.getElementById('new-candidate-party').value;

if (!id || !name || !party) {

alert('Please fill in all fields');

return;

}

// Add new candidate

candidates.push({

id,

name,

party,

votes: 0

});

// Clear form

document.getElementById('new-candidate-id').value = '';

document.getElementById('new-candidate-name').value = '';

document.getElementById('new-candidate-party').value = '';

// Hide modal

addCandidateModal.classList.add('hidden');

// Refresh candidates list and results

if (voterDashboard.classList.contains('hidden')) {

renderResults();

} else {

renderCandidates();

}

alert('New candidate added successfully!');

}

function resetSystem() {

if (confirm('Are you sure you want to reset the voting system? This will clear all votes and cannot be undone.')) {

// Reset all votes

candidates.forEach(c => c.votes = 0);

voters.forEach(v => v.hasVoted = false);

// Refresh displays

renderVotersTable();

renderResults();

alert('Voting system has been reset. All votes have been cleared.');

}

}

function switchTab(clickedTab, tabId) {

// Update active tab

document.querySelectorAll('.tab').forEach(tab => {

tab.classList.remove('active');

});

clickedTab.classList.add('active');

// Show corresponding content

document.querySelectorAll('.tab-content').forEach(content => {

content.classList.remove('active');

});

document.getElementById(`${tabId}-tab`).classList.add('active');

// If switching to results tab, refresh the results

if (tabId === 'results') {

renderResults();

}

}

function logout() {

currentUser = null;

voterDashboard.classList.add('hidden');

adminDashboard.classList.add('hidden');

loginSection.classList.remove('hidden');

// Clear login form

document.getElementById('user-id').value = '';

document.getElementById('password').value = '';

}

</script>

</body>

</html>